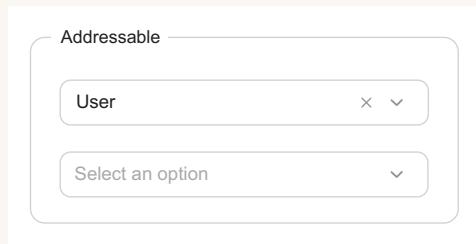


Available Fields

namespace where
all form fields
can be found
use Filament\Form\Components

| | | | |
|--------------|---|---|--|
| | Amount | | |
| TextInput | <input type="text" value="\$ 2000"/> | | |
| | The total amount | | |
| Checkbox | <input type="checkbox"/> Published | Published <input type="checkbox"/> | |
| Toggle | <input type="button" value="Is admin"/> | <input type="button" value="Is admin"/> | <input type="button" value="Is admin"/> |
| CheckboxList | <p>Planets</p> <p>Select all</p> <p><input type="checkbox"/> Mercury</p> <p><input type="checkbox"/> Venus</p> <p><input type="checkbox"/> Earth</p> <p><input type="checkbox"/> Mars</p> | <p>Planets</p> <p><input checked="" type="checkbox"/> Mercury</p> <p><input checked="" type="checkbox"/> Earth</p> <p><input type="checkbox"/> Venus</p> <p><input type="checkbox"/> Mars</p> | |
| Radio | <p>Like this post?</p> <p><input checked="" type="radio"/> Yes</p> <p><input type="radio"/> No</p> | <p>Like this post?</p> <p><input checked="" type="radio"/> Yes</p> <p><input type="radio"/> No</p> | <p>Like this post?</p> <p><input checked="" type="radio"/> Yes</p> <p><input type="radio"/> No</p> |
| ColorPicker | Color | <input type="text" value="#FF80C0"/> |  |
| Textarea | Description | <p>Learn how to use Filament components to write awesome applications</p> | |
| TagsInput | Tags | <input type="text" value="New tag"/> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> Tailwind CSS Alpine.js </div> | |
| Select | Status | Domain | |
| | <input type="text" value="Draft"/> | <input type="text" value="filamentphp"/> |  |

MorphToSelect



KeyValue

| Meta | | |
|-------------------------|--------------|--|
| | Key | Value |
| ↑↓ | description | Filament is a collection of Laravel packages |
| ↑↓ | og:type | website |
| ↑↓ | og:site_name | Filament |
| Add row | | |

ToggleButtons

Status

Draft

Scheduled

Published

Like this post?

✓ Yes ✗ No

Technologies

Tailwind CSS

Laravel

Alpine.js

Laravel Livewire

DateTimePicker

Published at

🕒

At

🕒

Date of birth

🕒

Date

Starts 🕒 at midnight

Markdown

Content

B I § ♂ H “ ” </> ≡ ≡ ≡ ≡ ≡ ≡ ≡ ≡

RichEditor

Content

B I § ♂ H Heading Subheading “ ” </> ≡ ≡ ≡ ≡ ≡ ≡ ≡ ≡

Upload

Attachment

Drag & Drop your files or [Browse](#)

ColorPicker Field

use `Filament\Forms\Components\ColorPicker`;

pick a color in a range of formats

`ColorPicker::make('color')`

Color

#FF80C0



Methods

- >hex()
- >hsl()
- >rgb()
- >rgba()



set the color format. Default: hex()

Textarea Field

use `Filament\Forms\Components\Textarea`;

interact with a multi-line string

`Textarea::make('description')`

Description

Learn how to use Filament components to write awesome applications



Methods

- >autosize() auto resize to fit the content
- >rows(10) textarea rows
- >cols(20) textarea columns
- >readOnly() make textarea read only

Validation specific to Textarea

- >minLength(2)
- >maxLength(1024)
- >length(100)

TagsInput Field

use `Filament\Forms\Components\TagsInput`;

also supports `spatie/laravel-tags`

interact with a list of tags

Tags

Tailwind CSS

Alpine.js

by default, tags are stored in JSON

Methods

- >separator(',') store tags in a separated string, instead of JSON
- >splitKeys(['Tab', ' ']) map buttons to create a new tag. By default, pressing "Enter" button, creates a new tag
- >tagSuffix('%') add prefix and suffix to tags without modifying the real state of the field. Useful to show presentational formatting to users without saving it
- >tagPrefix('%') presentational formatting to users without saving it
- >reorderable() allow the user to reorder tags
- >color('danger') change tag color. danger, gray, info, primary, success or warning

Validation specific to TagsInput

`TagsInput::make('tags')`

- >nestedRecursiveRules([
 'max:255',
])

← add validation rules
for each tag by passing
an array of rules



Add an array or json cast to the model property when using Eloquent

Utility Injection (Closure Customization)

Available utilities

| Parameter | Namespace |
|-----------------------|---|
| Component \$component | use Filament\Forms\Components\Component; |
| Livewire \$livewire | use Livewire\Component as Livewire; |
| ?Model \$record | use Illuminate\Database\Eloquent\Model; |
| Get \$get | use Filament\Forms\Get; |
| Set \$set | use Filament\Forms\Set; |
| \$state | - |
| \$old | old value of the field before it was updated (used with afterStateUpdate() method) |
| \$search | current search query (used in modifyQueryUsing on relationship()) |
| \$operation | current form operation (edit, create, view) |

in standalone form, the form's operation can be set using the `operation()` method on the form definition

Closure with utility injection

```
use Filament\Forms\Components\Component;

function (Component $component) {
    // ...
}
```

Closure with multiple utility injection

```
use Livewire\Component as Livewire;
use Filament\Forms\Get;

function (Livewire $livewire, Get $get, Set $set) {
    // ...
}
```

utility injection
(any order can be used)

Example: form action with utility injection

```
TextInput::make('cost')
    ->prefix('€')
    ->suffixAction(
        Action::make('copyCostToPrice')
            ->icon('heroicon-m-clipboard')
            ->requiresConfirmation()
            ->action(function (Set $set, $state) {
                $set('price', $state);
            })
    )
)
```

utility injection

Closure with Laravel's dependencies injection

```
use Illuminate\Http\Request;
any dependencies from Laravel's
container can also be injected

function (Request $request, Set $set) {
    // ...
}
```